ระบบปฏิบัติการ บทที่ 1

บทบาทหสำคัญของระบบปฏิบัติการในระบบคอมพิวเตอร์

การจัดการทรัพยากร การจัดการกระบวนการ การจัดการระบบไฟล์ และการจัดการเครือข่าย การจัดการความปลอดภัย เพื่อให้คอมพิวเตอร์ได้อย่างเป็นระบบและมีประสิทธิภาพ

เทคนิค Spooling

เทคนิค Spooling มีกระบวนการทำงานดังนี้

1. การจัดเก็บข้อมูล: ข้อมูลถูกเก็บในพื้นที่เก็บข้อมูลชั่วคราวในหน่วยความจำของระบบโดยไม่ต้องรอให้อุปกรณ์ส่วนย่อยพร้อมรับข้อมูล
2. การจัดการคิว:ข้อมูลที่จัดเก็บในพื้นที่เก็บข้อมูลชั่วคราวจะถูกจัดเก็บในลำดับของคิว เพื่อให้สามารถประมวลผลได้อย่างเป็นลำดับ
3. การแปลงข้อมูล:ข้อมูลที่ถูกจัดเก็บในพื้นที่เก็บข้อมูลชั่วคราวจะถูกแปลงรูปแบบให้เข้ากับรูปแบบของอุปกรณ์ส่วนย่อยที่กำลังรอการประมวลผล ซึ่งระบบปฏิบัติการจะดำเนินการแปลงรูปแบบนี้ให้อัตโนมัติ
4. การส่งข้อมูล: ข้อมูลที่ถูกแปลงรูปแบบให้เข้ากับอุปกรณ์ส่วนย่อยที่กำลังรอการประมวลผลจะถูกส่งไปยังอุปกรณ์ส่วนย่อยเมื่ออุปกรณ์พร้อมที่จะรับข้อมูลและประมวลผล

จุดเด่นของระบบปฏิบัติการMultiprogramming

มีการทำงานของหลายๆ โปรแกรมพร้อมกันบนคอมพิวเตอร์ โดยทำงานอย่างเป็นกระบวนการ จัดการทรัพยากรให้เป็นระเบียบและมีประสิทธิภาพ มีความปลอดภัย

คุณลักษณะของระบบปฏิบัติการต่อไปนี้

1) ระบบ Batch

ระบบที่ทำงานโดยการรวบรวมและประมวลผลงานที่เป็นกลุ่มเข้าด้วยกัน โดยไม่ต้องมีการตอบสนองทันทีต่อผู้ใช้งาน หากข้อผิดพลาดเกิดขึ้นระบบจะแก้ได้อย่างอัตโนมัติ โดยไม่ต้องรอการตอบสนองจากผู้ใช้งาน

2) ระบบ Multiprogramming

มีการประมวลผลแบบพร้อมกัน สามารถแบ่งเวลาในการประมวลผลระหว่างโปรแกรมต่างๆ ทำงานได้อย่างถูกต้อง มีการจัดการทรัพยากรให้เป็นระเบียบและมีประสิทธิภาพ และผู้ใช้งานสามารถทำงานได้โดยไม่ต้องรอการตอบสนองจากระบบ

3) ระบบแบ่งเวลา Time-sharing

ตอบ แบ่งเวลาการใช้ทรัพยากรของระบบคอมพิวเตอร์ระหว่างโปรแกรมหลายๆ โปรแกรมทำงาน โดยให้แต่ละโปรแกรมได้เวลาใช้ทรัพยากรในสัดส่วนที่กำหนด สามารถสลับงานไปมาระหว่างโปรแกรมต่างๆ โดยมีตัวกำหนดเวลาที่แน่นอน ทำให้ผู้ใช้งานสามารถทำงานพร้อมกันได้

4) ระบบ Multiprocessor

มีการทำงานของหลายๆ กระบวนการ (process) หรือโปรแกรมพร้อมกันบนหลายๆ ตัวประมวลผล (processor) สามารถแบ่งงานหรือโปรแกรมให้ทำงานพร้อมกันบนตัวประมวลผลหลายๆ ตัวได้ ทำให้เกิดการกระจายงานและเพิ่มประสิทธิภาพในการทำงาน สามารถ จัดการทรัพยากรร่วมกันให้เป็นระเบียบและมีประสิทธิภาพ ทำให้ทรัพยากรต่างๆ เช่น หน่วยความจำ อุปกรณ์เก็บข้อมูล เครือข่าย เป็นทรัพยากรที่สามารถใช้ร่วมกันได้อย่างมีประสิทธิภาพ

ระบบปฏิบัติการ บทที่ 2

ขั้นตอนหลักในการประมวลผลโปรแกรมและวงรอบการประมวลผล

การประมวลผลโปรแกรม จะมีขั้นตอนหลักดังนี้:

1. รับข้อมูลเข้า : โปรแกรมจะรับข้อมูลเข้ามาจากผู้ใช้ข้อมูลที่ต้องการให้โปรแกรมประมวลผล
2. ประมวลผล: ประมวลผลข้อมูลที่ได้รับเข้ามา ขั้นตอนนี้ใช้คำสั่งและอัลกอริทึมที่เพื่อดำเนินการ
3. ส่งออกผลลัพธ์: เมื่อข้อมูลถูกประมวลผลแล้ว โปรแกรมจะส่งออกผลลัพธ์ เช่น แสดงผลทางหน้าจอ

วงรอบการประมวลผล (Execution Cycle)วงรอบการประมวลผลประกอบด้วยการรับข้อมูลเข้า, กระบวนการประมวลผล, และส่งออกผลลัพธ์ ซึ่งจะเกิดขึ้นซ้ำๆ ไปเรื่อยๆ ตามที่โปรแกรมมีการทำงาน

แนวคิดหลักของกลไกการขัดจังหวะ

การจัดการเวลาและการทำงานของคอมพิวเตอร์ให้สลับสับเปลี่ยนและทำงานพร้อมกันเพื่อให้คอมพิวเตอร์ทำงานได้รวดเร็วและเรียบเนียนต่อเนื่อง ทำให้คอมพิวเตอร์ทำงานได้มากที่สุดตลอดเวลา

กลไก DMA จึงเหมาะกับอุปกรณ์อินพุต/เอาต์พุตที่มีความเร็วสูง

การถ่ายโอนข้อมูลระหว่างอุปกรณ์และหน่วยความจำหลักโดยตรง โดยไม่ต้องผ่านหน่วยประมวลผล (CPU) ทำให้มีประสิทธิภาพในการทำงานสูงและลดความหนักที่หน่วยประมวลผล

"การอ่านสัญญาณนาฬิกา" เกี่ยวข้องกับควบคุมเวลาของระบบคอมพิวเตอร์ และเป็นคำสั่งที่ใช้ในภาษาประสาทแบบ x86 เพื่อนำไปใช้ในการวัดประสิทธิภาพของโค้ดหรือการทดสอบความเร็วของระบบ

"การปิดสัญญาณการขัดจังหวะ" การปิดสัญญาณการขัดจังหวะเป็นเรื่องที่เกี่ยวข้องกับการควบคุมขั้นตอนการประมวลผลของหน่วยประมวลผล (CPU) และมีหลายวิธีการที่เกี่ยวข้องกับการประมวลผลและการจัดทำเครื่องรางของคอมพิวเตอร์

"การเปิดไฟล์" การเปิดไฟล์เป็นกระบวนการการเข้าถึงข้อมูลในไฟล์ของระบบคอมพิวเตอร์ ซึ่งมักใช้คำสั่งหรือฟังก์ชันที่เป็นส่วนหนึ่งของระบบปฏิบัติการเพื่อเปิดและอ่านข้อมูลในไฟล์

"การสลับโหมดจากโหมดผู้ใช้เป็นโหมดมอนิเตอร์" การสั่งการหรือเปลี่ยนโหมดการทำงานของระบบ ซึ่งมักมีการเปลี่ยนแปลงในเวลาที่ระบบคอมพิวเตอร์ถูกเปิดหรือเมื่อมีคำสั่งหรือฟังก์ชันที่เรียกใช้ในการสลับโหมด

"การสั่งเปิด/ปิด Timer"การตั้งค่าหรือควบคุมของอุปกรณ์ต่างๆ ในระบบคอมพิวเตอร์ เช่น การตั้งค่าการทำงานของตัวจับเวลาหรือตัวนับเวลาในระบบคอมพิวเตอร์

ระบบปฏิบัติการ บทที่ 3

ความแตกต่างของโพรเซสและโปรแกรม

โปรแกรมเป็นชุดคำสั่งที่เป็น Source Code เพื่อให้คอมพิวเตอร์ทำงานแต่โพรเซสจะมีการเก็บค่าของรีจิสเตอร์

Process Control Block

เป็นแหล่งเก็บข้อมูลของโพรเซส ประกอบไปด้วย Process state, Program counter ,CPU register ,CPU scheduling information ,Memory management information,Account information ,I/O status information

มีความสำคัญดังนี้

1. การจัดการโพรเซส: PCB จะเก็บข้อมูลที่ใช้ในการควบคุมทรัพยากรต่างๆ ในระบบ
2. การทำงานแบบพร้อมกัน: PCB ระบบปฏิบัติการสามารถสลับการทำงานของโพรเซสที่พร้อมทำงาน (Ready) หรือการถอดการทำงานออกจาก CPU ทำให้เกิดความเสถียรในการทำงานแบบพร้อมกันได้
3. การจัดสรรทรัพยากร: PCB เก็บข้อมูลเกี่ยวกับการใช้งานทรัพยากร เพื่อให้ระบบปฏิบัติการสามารถจัดสรรทรัพยากรให้เป็นไปอย่างมีประสิทธิภาพและเป็นระเบียบได้
4. การจัดการเวลา: PCB ประกอบด้วยข้อมูลเวลาที่โพรเซสเริ่มต้นการทำงาน (Creation Time) และเวลาที่โพรเซสหยุดการทำงาน (Termination Time) ใช้ในการติดตามและจัดการเวลาการทำงานของโพรเซส
5. การควบคุมการเข้าถึงทรัพยากร (Resource Access Control): PCB ควบคุมสิทธิ์การเข้าถึงทรัพยากร ที่โพรเซสใช้งาน เพื่อป้องกันปัญหาการแชร์ทรัพยากรที่ไม่เหมาะสมหรือขัดกัน
6. การวางกำหนดลำดับการทำงาน (Scheduling): ข้อมูลใน PCB เกี่ยวข้องกับลำดับความสำคัญ (Priority) และแนวทางการตั้งค่าต่างๆ เพื่อช่วยในการวางกำหนดลำดับการทำงานของโพรเซสในระบบ
7. ความเสถียร: PCB ช่วยให้ระบบปฏิบัติการทำงานได้อย่างเสถียร แม้ว่าจะเปลี่ยนแปลงสถานะของโพรเซส

การเปลี่ยนสถานะของโพรเซสขณะอยู่ในช่วงของการประมวลผล

1. การถูกเรียกใช้งานเมื่อโพรเซสถูกสร้างขึ้นหรือถูกเรียกใช้งาน โพรเซสจะถูกเปลี่ยนสถานะเป็น "Running"
2. การรอการเข้าถึงทรัพยากร โพรเซสจะถูกเปลี่ยนสถานะเป็น "Blocked"
3. การดำเนินการต่อ เมื่อเหตุการณ์ที่รออยู่ได้เกิดขึ้นและโพรเซสพร้อมที่จะดำเนินการต่อ โพรเซสที่อยู่ในสถานะ "Blocked" จะถูกเปลี่ยนสถานะเป็น "Running" เพื่อให้โพรเซสเริ่มทำงานต่อ
4. การเสร็จสิ้นการประมวลผล โพรเซสจะถูกเปลี่ยนสถานะเป็น "Terminated"
5. การถูกขัดจังหวะโพรเซสที่กำลังทำงานจะถูกเปลี่ยนสถานะเป็น "Ready" และโพรเซสที่ถูกขัดจังหวะจะถูกเปลี่ยนสถานะเป็น "Running"

ขั้นตอนของการทำ Context Switching

1. บันทึกสถานะปัจจุบันของโพรเซสที่กำลังทำงานอยู่ ซึ่งประกอบด้วยข้อมูลทั้งหมดของโพรเซส เช่น ค่าของ Register, ค่าของ Program Counter

2.เปลี่ยนสถานะโพรเซสที่กำลังทำงานอยู่จะถูกเปลี่ยนสถานะจาก "Running" เป็น "Blocked" หรือ "Ready"

3.เรียกโพรเซสใหม่ระบบจะเรียกโพรเซสใหม่ที่จะได้รับการทำงาน ซึ่งอาจเป็นโพรเซสที่พร้อมทำงาน (Ready) หรืออื่น ๆ ที่มีลำดับความสำคัญสูงขึ้น การเรียกโพรเซสใหม่จะทำให้โพรเซสใหม่เปลี่ยนสถานะเป็น "Running"

4. คืนค่าสถานะที่ถูกบันทึกไว้โพรเซสนี้จะเรียกคืนการทำงานของตัวเองได้ต่อไปเมื่อถูกเรียกใช้งานอีกครั้ง

เหตุผล Context Switching ถือว่าเป็นความสิ้นเปลืองของระบบ

1. เสียเวลาในการทำ Context Switching ต้องใช้เวลาในการบันทึกและกู้คืนสถานะของโพรเซส รวมถึงเรียกคืนการทำงานของโพรเซสใหม่ ทำให้เวลาที่ใช้ในการทำงานจริงๆ ลดลง

2. ทำให้เกิด Overhead มีความซับซ้อนในการจัดการโพรเซสและทรัพยากร ทำให้เกิดค่าใช้จ่ายในการทำงาน ทำให้ระบบทำงานช้าลง

3. การแบ่งเวลา (Time Slicing) ทำให้บางโพรเซสได้รับการทำงานน้อยลง

การสื่อสารระหว่างโปรเซส

โพรเซสสื่อสารหรือทำงานร่วมกันได้ผ่านกลไกที่เรียกว่า การแบ่งปันข้อมูลมีดังนี้

1. แบ่งปันหน่วยความจำ 2. การส่งข้อความ 3. การแชร์ไฟล์

ประโยชน์ของการใช้ระบบ Thread ในการประมวลผลโปรแกรมประยุกต์

ข้อดีของการใช้ Thread คือทำให้ระบบใช้หน่วยประมวลผลได้อย่างมีประสิทธิภาพเนื่องจาก Thread แต่ละ Thread จะแทนงานแต่ละอย่างยกตัวอย่างเช่น โปรแกรม Web browser จะมี Thread หนึ่งเพื่อแสดงรูปภาพหรือข้อความอีก Thread หนึ่งจะใช้เพื่อดึงข้อมูลจาก อินเตอร์เน็ต **ดังนั้น**เพื่อให้การทำงานต่าง ๆในโปรแกรมสำเร็จด้วยดี จึงมีการประมวลผลทุก Thread ในโพรเซส เพียงตัวโพรเซสเดียวเท่านั้นเนื่องจากเราแบ่งงานย่อยออกเป็นหลาย Thread ในโพรเซส แล้วแทนที่จะต้องมีการสลับโพรเซสดัง เช่น ในระบบแบบ Thread เดียว ดังนั้นจึงลดค่าใช้จ่าย

แบบฝึกหัดบทที่ 4

1. จงอธิบายความแตกต่างของการจัดลำดับการเข้าใช้ หน่วยประมวลผลแบบบังคับ(Preemptive scheduling) และแบบไม่บังคับ (Non-preemptive scheduling)

ตอบ ในระบบ Preemptive Scheduling โพรเซสที่กำลังทำงานอาจถูกหยุดในระหว่างที่มีการทำงานเพื่อให้โพรเซสอื่นที่มีลำดับความสำคัญสูงกว่าได้รับการใช้งาน แต่ในระบบ Non-preemptive Scheduling เมื่อโพรเซสได้รับการใช้งานแล้ว จะไม่มีการยกเลิกหรือขัดจังหวะโพรเซสนั้นในระหว่างการทำงาน โพรเซสจะทำงานจนกว่าจะเสร็จสิ้นหรือจนกว่าจะมีเหตุการณ์ที่ทำให้สลับไปใช้โพรเซสอื่น

2. พิจารณาเซทของโพรเซส พร้อมเวลาที่แต่ละโพรเซสต้องการใช้หน่วยประมวลผลและค่าระดับความสำคัญดังแสดงในตารางต่อไปนี้(ค่าระดับความสำคัญสูง หมายถึงมีความสำคัญมาก)

โพรเซส	เวลาที่ต้องการใช้หน่วยประมวลผล (มิลลิวินาที)	ค่าระดับความสำคัญ
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

สมมุติให้ทุกโพรเซสเข้ามาในแถวคอยเตรียมพร้อมเมื่อเวลา 0 โดยมีลำดับการเข้ามาก่อนหลังคือ P1, P2, P3, P4 และตามด้วย P5

2.1 จงวาด Gantt chart เพื่อแสดงลำดับการเข้าประมวลผลของโพรเซสโดยใช้ อัลกอริทึมแบบ FCFS, SJF, non-preemptive priority และRR (กำหนดค่า Time quantum = 1 มิลลิวินาที)

FCFS มาก่อน เข้าก่อน

P1	P2	P3	P4	P5
0	10	11	13	14	19

SJF เวลาน้อย เข้าก่อน

P2	P4	P3	P5	P1
0	1	2	4	9	19

NON-PREEMPTIVE PRIORITY ค่าระดับความสำคัญเข้าก่อน

P4	P1	P3	P5	P2
0	1	11	13	18	19

RR มาก่อนเข้าก่อน คล้ายๆ FCFS แต่ใช้ตามจำนวน Time quantum

P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P5	P5	P5	P5	P5
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

2.2 จงคำนวณหาค่าเวลาการรอคอยของแต่ละโพรเซสสำหรับการจัดลำดับด้วยอัลกอริทึมในข้อ 2.1

ตอบ

PROCESS	FCFS	SJF	NON-PREEMPTIVE PRIORITY	RR
P1	0	9	(11-1)=1	19-10=9
P2	10	0	(19-1)=18	2-1=1
P3	11	2	(13-2)=11	7-2=5
P4	13	1	(1-1)=0	4-1=3
P5	14	4	(18-5)=13	14-5=9
Average	9.6	3.2	8.6	5.4

2.3 จงคำนวณหาค่า Turnaround time ของแต่ละโพรเซสสำหรับการจัดลำดับด้วยอัลกอริทึมในข้อ 2.1

ตอบ

PROCESS	FCFS	SJF	NON-PREEMPTIVE PRIORITY	RR
P1	0+10=10	9+10=19	1+10=11	9+10=19
P2	10+1=11	0+1=1	18+1=19	1+1=2
P3	11+2=13	2+2=4	11+2=13	5+2=7
P4	13+1=14	1+1=2	0+1=1	3+1=4
P5	14+5=19	4+5=9	13+5=18	9+5=14
Average	13.4	7	12.4	9.2

2.4 จงหาว่าการจัดลำดับด้วยอัลกอริทึมใดมีค่าเฉลี่ยเวลาการรอคอยน้อยที่สุด

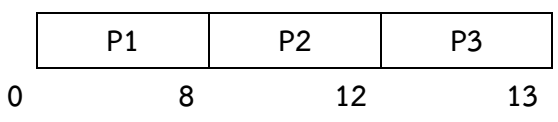
ตอบ SJF Average = 3.2 ms

3. สมมติให้โพรเซสต่อไปนี้เข้ามาที่แถวคอยเตรียมพร้อมเมื่อเวลาที่ต่างกัน และมีความต้องการใช้หน่วยประมวลผลตามเวลาที่ระบุในตารางต่อไปนี้

โพรเซส	เวลาที่เข้ามาในแถวคอยเตรียมพร้อม (มิลลิวินาที)	เวลาที่ต้องการใช้หน่วยประมวลผล (มิลลิวินาที)
P1	0.0	8
P2	0.4	4
P3	1.0	1

3.1 จงคำนวณหาค่าเฉลี่ย Turnaround time สำหรับการจัดลำดับด้วยอัลกอริทึมแบบ FCFS

FCFS



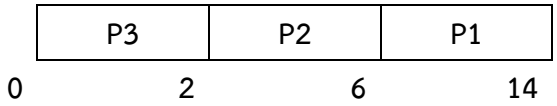
COMPLTion - ARAIVAL

PROCESS	*COMPLETION TIME*	*ARRIVAL TIME*	*TURNAROUND TIME*
P1	8.0	0.0	8
P2	12.0	0.4	11.6
P3	13.0	1.0	12
		Average	10.53

ตอบ ค่าเฉลี่ย Turnaround time สำหรับการจัดลำดับด้วยอัลกอริทึมแบบ FCFS คือ 10.53

3.2 จงคำนวณหาค่าเฉลี่ย Turnaround time สำหรับการจัดลำดับด้วยอัลกอริทึมแบบ Non preemptive SJF

SJF



PROCESS	*COMPLETION TIME*	*ARRIVAL TIME*	*TURNAROUND TIME*
P1	14	0.0	14
P2	6	0.4	5.6
P3	2	1.0	1
		Average	6.53

ตอบ ค่าเฉลี่ย Turnaround time สำหรับการจัดลำดับด้วยอัลกอริทึมแบบ Non preemptive SJF คือ 6.53

3.3 จงคำนวณหาค่าเฉลี่ย Turnaround time สำหรับการจัดลำดับด้วยอัลกอริทึมแบบ Preemptive SJF

SJF

PROCESS	*COMPLETION TIME*	*ARRIVAL TIME*	*TURNAROUND TIME*
P1	7	0.0	7
P2	6	0.4	5.6
P3	2	1.0	1
		Average	4.87

ตอบ ค่าเฉลี่ย Turnaround time สำหรับการจัดลำดับด้วยอัลกอริทึมแบบ Preemptive SJF คือ 4.87

ระบบปฏิบัติการ บทที่ 5

Busy waiting และ Busy waiting มีบทบาทอย่างไรในการสื่อสารระหว่างโพรเซส

Busy waiting มีบทบาทในการรอการเปลี่ยนแปลงของข้อมูลหรือสถานะที่ถูกส่งผ่านการแบ่งปัน

ข้อกำหนดในการแก้ปัญหาการเข้าใช้ส่วนวิกฤต

การแก้ปัญหาเงื่อนไขการแข่งขันที่เกิดจากการใช้ส่วนวิกฤติร่วมกันของโพรเซส

1) Mutual exclusion : กล่าวคือต้องมีโพรเซสเดียวเท่านั้นที่เข้าใช้ส่วนวิกฤติ ณ ขณะเวลาใดๆ ซึ่งถ้าโพรเซสแรกยังประมวลผลในส่วนวิกฤติไม่แล้วเสร็จ โพรเซสอื่นจะเข้ามาใช้ส่วนวิกฤติไม่ได้

2) Progress : ถ้าไม่มีโพรเซสใดๆ ใช้ส่วนวิกฤติหรือมีโพรเซสตั้งแต่1โพรเซสขึ้นไปต้องการที่จะเข้าใช้ส่วนวิกฤตินั้น โพรเซสเหล่านั้นจะต้องผ่านขบวนการคัดเลือกโดยไม่มีการกีดกันตัวเอง หรือกีดกันระหว่างกัน

3) Bounded waiting : เวลาที่โพรเซสใดๆ ไม่มีการรอแบบไม่มีที่สิ้นสุด

Semaphore เพื่อแก้ปัญหาการใช้ส่วนวิกฤติร่วมกันของโพรเซส

1) เริ่มต้นกำหนดให้ตัวแปร Semaphore มีค่า >= 0

2) สำหรับตัวดำเนินการ wait จะทำการลดค่าของตัวแปร Semaphore ลงครั้งละ 1 ถ้าลดค่าลงจนตัวแปร Semaphore มีค่า < 0 แล้วโพรเซสนั้นจะถูกสกัดกั้น

3) สำหรับตัวดำเนินการ signal จะเป็นการเพิ่มค่าของตัวแปร Semaphore ขึ้นครั้งละ 1 ถ้าค่าตัวแปร Semaphore ยังคง <= 0 แล้วโพรเซสที่ถูกสกัดกั้นไว้นั้นจะถูกปลดปล่อย

ระบบปฏิบัติการ บทที่ 6

วิธีการป้องกันการเกิด Deadlock

1 **Mutual Exclusion** ทรัพยากรแต่ละชนิดอนุญาตให้โพรเซสใดๆ ใช้ร่วมกันได้ เช่น แฟ้มข้อมูลแบบอ่านอย่างเดียว ทรัพยากรบางชนิดชนิดไม่อนุญาตให้โพรเซสใช้ร่วมกัน

2 **Hold and Wait** การดำเนินการได้ตามข้อตกลงข้อใดข้อหนึ่งต่อไปนี้

1) ถ้าโพรเซสใดต้องการใช้ทรัพยากร โพรเซสนั้นต้องได้ถือครองทรัพยากรที่ต้องการทั้งหมด หรือไม่ได้ถือครอง

2) ระบบปฏิบัติการอาจจะอนุญาตให้โพรเซสถือครองทรัพยากรใหม่ได้ก็ต่อเมื่อโพรเซสนั้นไม่มี ทรัพยากรที่ถือครองเหลืออยู่ แต่ถ้ามีโพรเซสนั้นต้องคืนทรัพยากรก่อนแล้วจึงจะทำการขอ ทรัพยากรใหม่ได้

3. **No Preemption** มีการบังคับให้โพรเซสใดๆ คืนทรัพยากรได้ ซึ่งวิธีการนี้จะใช้ได้กับการถือครองทรัพยากรแบบบังคับ (Preemptive resource) ส่วน ทรัพยากรแบบบังคับคืนไม่ได้ (Non-preemptive resource) ระบบปฏิบัติการจะไม่สามารถบังคับให้โพรเซสที่กำลังถือครองทรัพยากรแบบบังคับคืนไม่ได้ทำการ กล่าวคือถ้าโพรเซสยึดครอง ทรัพยากรบางส่วนและต้องการใช้ทรัพยากรเพิ่มเติม

4. Circular Wait ระบบปฏิบัติการ ต้องไม่อนุญาตให้เกิดสภาวะการรอคอยอย่างเป็นวงรอบ

การกู้ระบบคืนจากสถานะ Deadlock

มีหลักการคือต้องทำให้วงรอบการเกิด Deadlock แตก เพื่อทำการย้อนสถานะของโพรเซสใด ๆ กลับไปก่อนที่จะเกิดสภาวะ Deadlock ซึ่งการทำให้วงรอบ Deadlock แตกสามารถทำได้ 2 วิธี คือ

1) การหยุดการทำงานของโพรเซส (Process Termination) ระบบปฏิบัติการต้องทำการหยุดทุกโพรเซสที่อยู่ในวง Deadlock ลง แต่มักจะมีค่าใช้จ่ายสูง

2) การบังคับคืนทรัพยากร (Resource Preemption) วิธีการนี้จะเป็นการบังคับคืนทรัพยากรจากโพรเซสและจัดสรรให้กับโพรเซสอื่นจนกระทั่ง วงรอบ Deadlock แตกลง ซึ่งวิธีการนี้ก็มีประเด็นที่ต้องพิจารณาดังต่อไปนี้

• การเลือกโพรเซสที่ถูกบังคับให้คืนทรัพยากร โดยในที่นี้อาจพิจารณาจากจำนวนทรัพยากร ที่โพรเซสนั้นถือครองอยู่

• การย้อนกลับ คือ กรณีหลังจากที่เราบังคับให้โพรเซสคืนทรัพยากรและจัดสรรทรัพยากรนั้น ให้โพรเซสอื่นแทนแล้ว ระบบปฏิบัติการจะหยุดการทำงานของโพรเซสเหล่านั้นลง แล้วทำการ restart ใหม่โดยจะมีการกำหนดค่าหรือทรัพยากรต่างๆให้โพรเซสอีกครั้งแต่จะเป็นค่าก่อนที่จะมีการเกิดDeadlock

• ระบบจะรับประกันได้อย่างไรว่าเมื่อเกิด Deadlock ขึ้นจะไม่มีโพรเซสใดโพรเซสหนึ่งที่โดน บังคับให้คืนทรัพยากรอยู่เสมอๆ ขณะที่โพรเซสอื่นไม่โดน

ระบบปฏิบัติการ บทที่ 7

ความแตกต่างของ Logical address และPhysical address

Logical address ตำแหน่งของคำสั่งหรือตัวแปรที่ถูกสร้างขึ้นและอ้างอิงโดยหน่วยประมวลผลแต่ Physical address ตำแหน่งของคำสั่งหรือตัวแปรที่อยู่ในหน่วยความจำหลักจริง

Logical address ที่มีจำนวน Page 64 Page แต่ละ Page มีขนาด 1024 byte ซึ่ง Logical address นี้จะถูกแมพไปเป็น Physical address ขนาด 32 Frame

หาว่าต้องใช้กี่บิตเพื่อแทน Logical address

ตอบ 640 บิต หาได้จากขนาดของแต่ละ Page =10 บิต (เพราะ 2^10=1024)

จำนวน Page = 64

หาว่าต้องใช้กี่บิตเพื่อแทน Physical address

ใช้ 15 บิตหาได้จากFrame มีขนาด 1024 byte(2^10บิต)

จำนวนบิตที่ใช้=log2(ขนาดของแต่ละ Frame)+log2(จำนวนFrame)

=log2(2^10)+log2(32)=10+5=15บิต

การเกิด Internal fragmentation และการเกิด External fragmentation

Internal Fragmentation เกิดขึ้นเมื่อมีพื้นที่ว่างภายในบล็อกหรือส่วนที่ถูกจองขึ้นมาแต่ไม่ได้ถูกใช้งานทั้งหมด ซึ่งส่งผลให้มีพื้นที่ว่างที่ไม่สามารถนำมาใช้งานได้

ตัวอย่างเช่น ถ้ามีบล็อกขนาด 4096 byte (4 KB) และมีข้อมูลที่ใช้งานจริงเพียง 3000 byte จะมี Internal Fragmentation อยู่ที่ 1096 byte (4 KB - 3000 byte).

External Fragmentation เกิดขึ้นเมื่อมีพื้นที่ว่างหลายๆ ส่วนที่กระจายอยู่รอบๆ ในพื้นที่ทั้งหมด ทำให้ไม่สามารถจัดให้มีพื้นที่ใหญ่พอที่จะให้โปรแกรมทำงานต่อไปได้

ตัวอย่างเช่น มีพื้นที่ว่างทั้งหมด 16 KB แต่มีช่องว่างที่ใช้งานได้แค่ 4 KB, 4 KB, 4 KB, และ 2 KB ตามลำดับ ทำให้ไม่สามารถให้โปรแกรมที่ต้องการใช้พื้นที่ 8 KB ได้

ความแตกต่างระหว่าง Internal Fragmentation และ External Fragmentation

Internal Fragmentation เกิดจากการให้พื้นที่มากเกินไปสำหรับข้อมูลที่ถูกจัดเก็บส่งผลให้มีพื้นที่ว่างที่ไม่สามารถนำมาใช้งานได้ แต่External Fragmentation เกิดจากการกระจายของพื้นที่ที่เป็นช่องว่าง Internal Fragmentation เกิดภายในบล็อกหรือส่วนที่ถูกจองแต่External Fragmentation เกิดในพื้นที่ทั้งหมดที่ยังไม่ถูกจองส่งผลให้ไม่สามารถจัดให้โปรแกรมทำงานต่อได้

พิจารณา Page Reference ต่อไปนี้

2 3 4 5 3 1 4 5 1 7 3 6 1 5 3

กำหนดให้จำนวนเฟรม เท่ากับ 3

จงหาจำนวน Page Faults , Success Function , Failure Function ของการสลับ Page โดยใช้อัลกอริทึมต่อไปนี้

4.1) FIFO

Page reference	2	3	4	5	3	1	4	5	1	7	3	6	1	5	3	
Number of		2	3	4	5											
Frame = 3			2	3	4											
				2	3											
Page fault		+	+	+												

-เพิ่ม page 2 3 4 เข้าใน memory ตามลำดับหลังจากนั้นตรวจสอบ pageต่อไป

-page ต่อไปหมายเลข 5 ตรวจสอบว่ามี 5 อยู่ใน memory หรือไม่ หากยังไม่มีให้เอา page 2 ออกการนำ page ออกจาก memory จะทำให้เกิด Page fault

Page reference	2	3	4	5	3	1	4	5	1	7	3	6	1	5	3	
Number of		2	3	4	5	5	1	1	1	1	7	3	6	1	5	3
Frame = 3			2	3	4	4	5	5	5	5	1	7	3	6	1	5
				2	3	3	4	4	4	4	5	1	7	3	6	1
Page fault		+	+	+		+				+	+	+	+	+	+	

-page ต่อไปหมายเลข 3 มี 3 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 1 ตรวจสอบว่ายังไม่มี 1 ใน memory โดยนำ page 3 ออกจาก memory และเพิ่ม page 1 ลงใน memory

-page ต่อไปหมายเลข 4 มี 4 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 5 มี 5 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 1 มี 1 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 7 ตรวจสอบว่ายังไม่มี 7 ใน memory โดยนำ page 4 ออกจาก memory และเพิ่ม page 7 ลงใน memory

-page ต่อไปหมายเลข 3 ตรวจสอบว่ายังไม่มี 3 ใน memory โดยนำ page 5 ออกจาก memory และเพิ่ม page 3 ลงใน memory

-page ต่อไปหมายเลข 6 ตรวจสอบว่ายังไม่มี 6 ใน memory โดยนำ page 1 ออกจาก memory และเพิ่ม page 6 ลงใน memory
-page ต่อไปหมายเลข 1 ตรวจสอบว่ายังไม่มี 1 ใน memory โดยนำ page 7 ออกจาก memory และเพิ่ม page 1 ลงใน memory
-page ต่อไปหมายเลข 5 ตรวจสอบว่ายังไม่มี 5 ใน memory โดยนำ page 3 ออกจาก memory และเพิ่ม page 5 ลงใน memory
-page ต่อไปหมายเลข 3 ตรวจสอบว่ายังไม่มี 3 ใน memory โดยนำ page 6 ออกจาก memory และเพิ่ม page 3 ลงใน memory
F = 10 S = 5 f = 10/15*100=66.66%

4.2) Optimal

Page reference	2	3	4	5	3	1	4	5	1	7	3	6	1	5	3	
Number of		2	3	4	4											
Frame = 3			2	3	3											
				2	5											
Page fault		+	+	+												

-เพิ่ม page 2 3 4 เข้าใน memory ตามลำดับหลังจากนั้นตรวจสอบ pageต่อไป

-page ต่อไปหมายเลข 5 ตรวจสอบว่ามี 5 อยู่ใน memory หรือไม่ หากยังไม่มีให้ดู page ที่มีโอกาสเรียกใช้อีกครั้งนานที่สุด ดังนั้นจึงนำ page 2 ออกจาก memory และแทน page 5 ในตำแหน่งนั้น จะทำให้เกิด Page fault

Page reference	2	3	4	5	3	1	4	5	1	7	3	6	1	5	3	
Number of		2	3	4	4	4	4	4	4	4	4	3	3	3	3	3
Frame = 3			2	3	3	3	1	1	1	1	1	1	1	1	5	5
				2	5	5	5	5	5	5	7	7	6	6	6	6
Page fault		+	+	+		+				+	+	+	+	+		

-page ต่อไปหมายเลข 3 มี 3 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 1 ตรวจสอบว่ายังไม่มี 1 ใน memory โดยนำ page ที่มีโอกาสเรียกใช้อีกครั้งนานที่สุดออก ดังนั้นจึงนำ page 3 ออกจาก memory และแทน page 1 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 4 มี 4 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 5 มี 5 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 1 มี 1 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 7 ตรวจสอบว่ายังไม่มี 7 ใน memory โดยนำ page ที่มีโอกาสเรียกใช้อีกครั้งนานที่สุดออก ดังนั้นจึงนำ page 5 ออกจาก memory และแทน page 7 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 3 ตรวจสอบว่ายังไม่มี 3 ใน memory โดยนำ page ที่มีโอกาสเรียกใช้อีกครั้งนานที่สุดออก ดังนั้นจึงนำ page 4 ออกจาก memory และแทน page 3 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 6 ตรวจสอบว่ายังไม่มี 6 ใน memory โดยนำ page ที่มีโอกาสเรียกใช้อีกครั้งนานที่สุดออก ดังนั้นจึงนำ page 7 ออกจาก memory และแทน page 6 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 1 มี 1 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 5 ตรวจสอบว่ายังไม่มี 5 ใน memory โดยนำ page ที่มีโอกาสเรียกใช้อีกครั้งนานที่สุดออก ดังนั้นจึงนำ page 1 ออกจาก memory และแทน page 5 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 3 มี 3 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

F =9 S = 6 f =9/15*100 =60%

4.3) LRU

Page reference	2	3	4	5	3	1	4	5	1	7	3	6	1	5	3
Number of		2	3	4	4										
Frame = 3			2	3	3										
				2	5										
Page fault		+	+	+											

-เพิ่ม page 2 3 4 เข้าใน memory ตามลำดับหลังจากนั้นตรวจสอบ pageต่อไป

-page ต่อไปหมายเลข 5 ตรวจสอบว่ามี 5 อยู่ใน memory หรือไม่ หากยังไม่มีให้ดู page ที่เคยถูกใช้มาแล้วนานทีสุด ดังนั้นจึงนำ page 2 ออกจาก memory และแทน page 5 ในตำแหน่งนั้น จะทำให้เกิด Page fault

Page reference	2	3	4	5	3	1	4	5	1	7	3	6	1	5	3
Number of		2	3	4	4	4	1	1	1	1	1	1	6	6	6	3
Frame = 3			2	3	3	3	3	3	5	5	5	3	3	3	5	5
				2	5	5	5	4	4	4	7	7	7	1	1	1
Page fault		+	+	+		+	+	+		+	+	+	+	+	+	

-page ต่อไปหมายเลข 3 มี 3 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 1 ตรวจสอบว่ายังไม่มี 1 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 4 ออกจาก memory และแทน page 1 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 4 ตรวจสอบว่ายังไม่มี 4 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 5 ออกจาก memory และแทน page 4 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 5 ตรวจสอบว่ายังไม่มี 5 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 3 ออกจาก memory และแทน page 5 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 1 มี 1 อยู่ใน memory แล้วก็ไม่ต้องสลับ page

-page ต่อไปหมายเลข 7 ตรวจสอบว่ายังไม่มี 7 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 4 ออกจาก memory และแทน page 7 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 3 ตรวจสอบว่ายังไม่มี 3 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 5 ออกจาก memory และแทน page 3 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 6 ตรวจสอบว่ายังไม่มี 6 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 1 ออกจาก memory และแทน page 6 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 1 ตรวจสอบว่ายังไม่มี 1 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 7 ออกจาก memory และแทน page 1 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 5 ตรวจสอบว่ายังไม่มี 5 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 3 ออกจาก memory และแทน page 5 ในตำแหน่งนั้น

-page ต่อไปหมายเลข 3 ตรวจสอบว่ายังไม่มี 3 ใน memory โดยนำ page เคยถูกใช้มาแล้วนานทีสุดออก ดังนั้นจึงนำ page 6 ออกจาก memory และแทน page 3 ในตำแหน่งนั้น

F = 12 S = 3 f = 12/15*100=80%

ระบบปฏิบัติการ บทที่ 8

คุณลักษณะของแฟ้มข้อมูล

• ชื่อ (Name) ชื่อแฟ้มข้อมูลที่เป็นสัญลักษณ์เป็นเพียงข้อมูลที่เก็บไว้ในรูปแบบที่อ่านได้โดยมนุษย์

• ตัวระบุ (Identifier) แท็กที่ไม่ซ้ำกันนี้มักจะเป็นหมายเลขตัวระบุแฟ้มข้อมูลภายในระบบแฟ้ม

• ประเภท (Type) ข้อมูลนี้จำเป็นสำหรับระบบที่สนับสนุนประเภทต่าง ๆ ของแฟ้มข้อมูล

• ตำแหน่ง (Location) ข้อมูลนี้เป็นตัวชี้ไปยังอุปกรณ์และตำแหน่งของแฟ้มที่อยู่ในฮาร์ดดิสก์

• ขนาด (Size) จะบอกขนาดปัจจุบันของแฟ้มข้อมูล (หน่วยเป็นไบต์) ซึ่งขนาดสูงสุดจะถูก กำหนดเอาไว้

• การป้องกัน (Protection) จะใช้ควบคุมสิทธิ์ในการเข้าถึงแฟ้มข้อมูลต่าง ๆ เช่น การอ่าน เขียนแฟ้มข้อมูล

• วันเวลาของผู้ใช้(Time, Date, and User Identification) ข้อมูลนี้จะถูกเก็บไว้เมื่อมีการ แก้ไขแฟ้มข้อมูลนี้ในครั้งล่าสุด ซึ่งมีประโยชน์สำหรับป้องกันความปลอดภัย

ประเภทของแฟ้มข้อมูลที่คุณรู้จักและบอกว่าประเภทของแฟ้มข้อมูล

- Text File คือ ไฟล์ข้อความใช้สำหรับเก็บข้อมูลที่เป็นข้อความหรือตัวอักษร
- Image Files คือ ไฟล์รูปภาพใช้สำหรับเก็บภาพและภาพเคลื่อนไหว

ข้อแตกต่างระหว่างไดเรกทอรี่แบบกราฟโดยทั่วไปกับแบบไดเรกทอรี่กราฟแบบไม่เป็นวงจร

ไดเรกทอรี่แบบกราฟทั่วไปนั้นอาจมีการเชื่อมโยงในรูปแบบวงจรทำให้มีโอกาสทำให้เกิดการลูปได้ ซึ่งสามารถก่อปัญหาและทำให้การนำทางในระบบเสียง่ายแต่ไดเรกทอรีกราฟแบบไม่เป็นวงจรไม่มีวงจร ทำให้ไม่มีการเชื่อมโยงกลับมาที่โหนดเดิม ซึ่งจะไม่มีโอกาสเกิดการลูป ทำให้การนำทางเป็นไปอย่างมีประสิทธิภาพ

ประเภทในการเข้าถึงแฟ้มข้อมูลโดยการควบคุมจำแนกสิทธิ์ของผู้ใช้ การป้องกันและกำหนดสิทธิ์ในระบบปฏิบัติการยูนิกส์

การควบคุมการเข้าถึงแฟ้มข้อมูลและการกำหนดสิทธิ์ในระบบปฏิบัติการยูนิกส์ (Unix-like) นั้นสามารถทำได้ผ่านการใช้งาน Permission และ Ownership ซึ่งมีลักษณะดังนี้:

1. Permission (สิทธิ์)

- Read (r):สิทธิ์ในการอ่าน (Read) ข้อมูลหรือไฟล์
- Write (w):สิทธิ์ในการเขียน (Write) หรือแก้ไขข้อมูลหรือไฟล์
- Execute (x):สิทธิ์ในการรัน (Execute) ไฟล์ (สำหรับไฟล์ที่เป็นโปรแกรมหรือสคริปต์)

Permission จะถูกกำหนดต่อไฟล์หรือไดเรกทอรีเพื่อกำหนดสิทธิ์ของผู้ใช้ดังนี้

- Owner (เจ้าของ):คนที่สร้างหรือเป็นเจ้าของไฟล์หรือไดเรกทอรี
- Group (กลุ่ม):คนที่อยู่ในกลุ่มของเจ้าของไฟล์หรือไดเรกทอรี

ระบบปฏิบัติการ บทที่ 8

ผู้ใช้ระบบทั้งหมดที่ไม่ใช่เจ้าของไฟล์หรือไดเรกทอรี

2. Ownership (การเป็นเจ้าของ)

- User (ผู้ใช้เจ้าของ):คนที่สร้างหรือเป็นเจ้าของไฟล์หรือไดเรกทอรี
- Group (กลุ่ม):กลุ่มผู้ใช้ที่มีสิทธิ์ในการเข้าถึงไฟล์หรือไดเรกทอรีนี้

ผู้ใช้ระบบทั้งหมดที่ไม่ใช่เจ้าของไฟล์หรือไดเรกทอรี

3. การกำหนดสิทธิ์:การกำหนดสิทธิ์นั้นใช้คำสั่ง chmod โดยมีรูปแบบที่สำคัญดังนี้

- chmod u=rwx,g=rx,o=r file_name:กำหนดสิทธิ์ให้กับเจ้าของ (User) มีสิทธิ์อ่าน เขียน และรัน กลุ่ม (Group) มีสิทธิ์อ่านและรัน และ public มีสิทธิ์อ่าน
- chmod 755 file_name:กำหนดสิทธิ์ให้กับเจ้าของ (User) มีสิทธิ์อ่าน เขียน และรัน กลุ่ม (Group) และ public มีสิทธิ์อ่านและรัน

ในระบบปฏิบัติการในปัจจุบันที่กำหนดแนวทางป้องกันการเข้าถึงแฟ้มข้อมูล และการเข้าถึงไดเรกทอรี่ และไดเรกทอรี่ย่อยว่าใช้หลักการอย่างไร โดยยกตัวอย่างมาอย่างน้อย 1 ระบบปฏิบัติการ

ตอบ **ระบบปฏิบัติการ macOS**

- Permission (สิทธิ์):macOS ใช้ Permission (สิทธิ์) แบบ Read (r), Write (w), Execute (x) และสิทธิ์สำหรับเจ้าของ (Owner), กลุ่ม (Group), และ Public (Others) เพื่อกำหนดการเข้าถึงแฟ้มข้อมูลและไดเรกทอรี
- Ownership (การเป็นเจ้าของ):macOS ใช้ User และ Group เพื่อระบุการเป็นเจ้าของแฟ้มข้อมูลและไดเรกทอรี